LED 点阵显示电路的设计

一、设计内容

1．用四块 8×8LED 点阵设计一个 16×16 点阵字符显示电路，动态扫描显示方式在显示技术中的应用。

2．掌握点阵显示的硬件接口及软件编程。

3．调用字库文件和 SJ8002C 驱动函数，编写在 LED 点阵上显示任意字符的程序。

二、8 × 8 的 LED 点阵显示器原理

1．8 × 8 的 LED 点阵显示器结构

8 × 8 的 LED 点阵显示器，是由 64 个 LED 组成，内部电路如图 12-4 所示。共阳极的 8 × 8 的 LED 点阵显示器的典型连接方式是：每一行的阳极连在一起，由行扫描码锁存器和驱动器的一位控制，总共 8 行阳极连线由 8 位分别控制；每一列的 8 个阴极连在一起，由字形行码锁存器和驱动器的一位控制，总共 8 列阴极连线由 8 位分别控制。

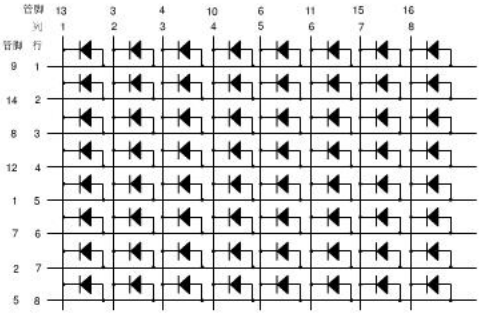


图 12-4　一个 8×8LED 点阵块的电路连接

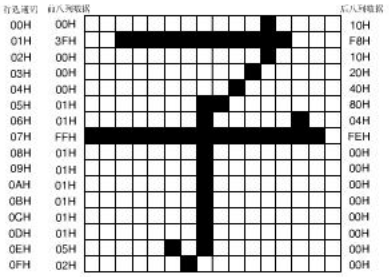


图 12-5 汉字“子”的字型点阵码图

2．点阵字符的字型码

在写点阵显示的驱动之前，应知道各显示字符的字形码。上图 12-5 是四块 8 × 8 的 LED 点阵拼成的一个 16 × 16 点阵（16 列 16 行）作为 1 位字符显示的点阵码图。（本实验也采用相同的显示方式显示一个字符）。下面列出“电子”2 个字符的点阵码。

“电”--- 02H 00H、02H 00H、02H 00H、02H 10H、7FH F8H、42H 10H、42H 10H、7FH F0H、42H 10H、42H 10H、7FH F0H、42H 10H、02H 04H、02H 04H、02H 04H、01H FCH；

“子”--- 00H 10H、3FH F8H、00H 10H、00H 20H、00H 40H、01H 80H、01H 04H、FFH FEH、01H 00H、01H 00H、01H 00H、01H 00H、01H 00H、01H 00H、05H 00H、02H 00H；

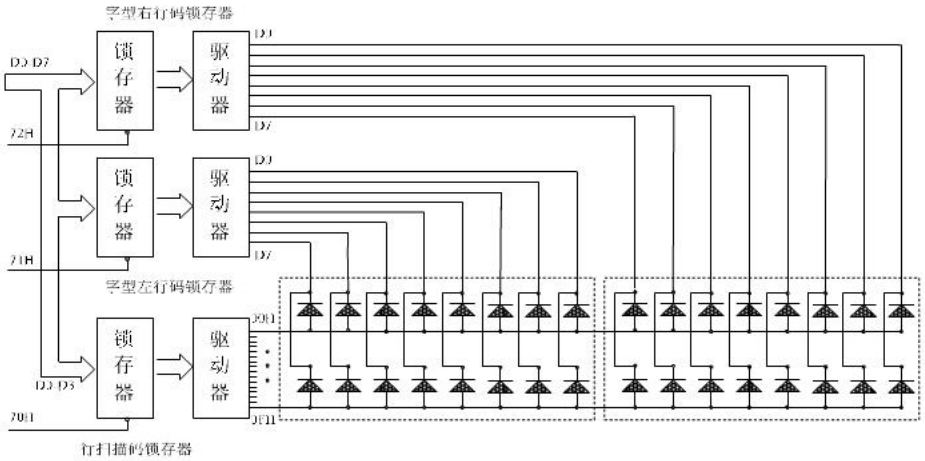
3．点阵字符的驱动

点阵式 LED 显示器采用逐行扫描式工作。要使点阵显示出一个的字符的编程方法是：首先向字形行扫描码锁存器输出行扫描码，选通第一行，同时向字符锁存器列写入该行的字型码，完成一行的扫描。然后，按相同的方式选通第二行，写第二行的字型码……由此类推，逐行扫描，直到写完所有行的字型码，完成一个字符的一遍扫描。

如果要使多个点阵循环显示多个字符，只要把显示的各个字符按顺序安排在显示缓冲区，然后根据显示的字符去查表，再按一定的时序向各个字形行码锁存器和行扫描器输入相应的字形行码和行扫描码，便可达到目的。

4．LED 点阵式显示器实验电路

本实验所用的点阵显示板是外接在实验平台并行 I/O 扩展插座上，所有的选通地址都来自实验平台上的 74LS138 地址译码电路。电路每个行的选通信号都加有三极管驱动，以增加电流的驱动能力。行扫描码锁存器的选通地址为 70H，行各行选通码为：00H~0FH。点阵码式 16 位的，因此需要 2 个 8 位的行码锁存器，分别控制字符的左右字型码。其地址分别为 71H(左)和 72H（右）。如下图所示。



LED 点阵显示器接口电路图

本设计共用到三个端口地址：

70H：输出口，行扫描码锁存器地址。

71H：输出口，字形行码左锁存器地址。

72H：输出口，字形行码右锁存器地址。

5．字库原理

LED 点阵显示的原理和过程都很简单，只是输入每个字符的显示点阵码很繁琐。多字符的显示采用的方法是读字库，查出显示字的点阵码，再显示到显示屏上。

国标对汉字库（区位码字库）的结构作了统一的规定：将汉字库分为若干个区，每个区有 94 个汉字。每一个汉字在字库中有一个固定的区和位，即每一个汉字有一个区位码。知道了区位码也就相当于知道了汉字在字库中的位置。由于汉字的内码与区位码有一定的关系，所以，只要通过内码就可以得到区位码，从而也就得到了汉字的字模。

由于计算机对西文字符采用一个字节表示，汉字用二个字节（GBK 内码）表示。为了保证中西文兼容，因此规定每个字节只用七位，若两个字节的最高位均为 1，则该字符为汉字。即计算机中的数字和一些特殊符号按 ASCⅡ编码方式，汉字和一些符号是 GBK 内码编码方式来表示。而点阵显示字库是按区位编码方式排列。因此，读字库之前，要完成 ASCⅡ编码与区位编码的转换和 GBK 内码编码与区位编码之间的转换

6．UCDOS 16 点字库文件（HZK16j）

本设计是读 UCDOS 16 点字库文件（HZK16j）。字库中的汉字按共阳极 16×16 点阵模式存储，即每个汉字由 16×16=256 个点组成，占用 16×2=32 个连续的字节单元。字节的每一位（bit）表示一个点的属性：1 表示亮点，0 表示暗点。字符点阵是按照汉字区位码排列的，连续的两个字节表示该汉字字模的一行。

（1）GBK 内码与相应区位码的转换

汉字的内码从一些图形字符开始，起始编码为 A1A1H；汉字内码第一位为区码，每区为 94 个汉字，第二位为位码。汉字点阵数据在字库中的偏移量为：

((区码-A1H)*94 + (位码-A1H))*32L

由此可编辑偏移量的源代码为：

```
inter_code.ed[0]=character[count];
count++;
inter_code.ed[1]=character[count];
point_quwei=inter_code.ed;
wei=((point_quwei & 0xff00) >>8 )- 0xa1;
```

```
qu=(point_quwei & 0x00ff) - 0xa1;
rec=(qu*94+wei)*32L;
```

（2）ASCⅡ码与相应区位码的转换

ASCⅡ码的所有符号全在区位码的第三区，位码的偏移量为 0x21H。ASCⅡ码在字库中的偏移量为：

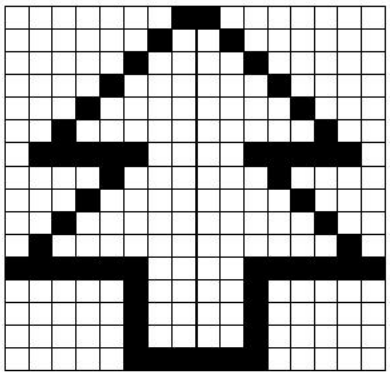
(区码*94 + (位码-21H))*32L

由此可编辑偏移量的源代码为：

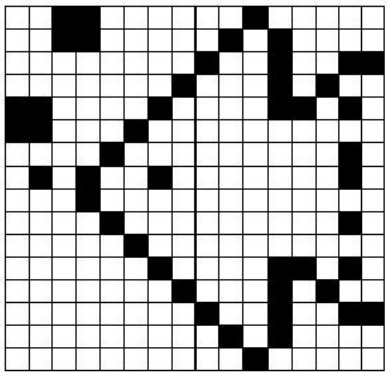
```
inter_code.ed[0]=character[count];
point_quwei=inter_code.ed;
wei= ((point_quwei & 0x00ff)-0x21;
qu=0x03-0x01;
rec=(qu*94+wei)*32L;
```

三、点阵显示图形

1．由 LED 点阵显示原理写出下图的点阵码。



树型图案



鱼吐气泡图案

2．编写显示程序段。

①打开 CVI 软件，创建一个空的工程文件。进入用户界面的编辑窗口，创建 UIR 文件，对用户界面进行设计。

参考设计界面如下图；控件的主要属性设置如下表

用户图形参考界面

主要控件的基本属性列表

	类型	Constant Name	Callback Function	作用
面板	Panel	PANEL		
控件	Command Button	START	start	显示启动
	Command Button	QUIT	Quit	退出程序
	Ring Slide	SPEED		显示刷新速度调节
	Decoration	DECORATION_1－4		美观

②在用户界面编辑窗口，由图形界面生成源代码框架，并给工程文件命名。

③向 C 代码文件添加 LED 点阵板显示程序段如下。(数组 unsigned char tree[32]={ }的元素为树型图案的点阵码，unsigned char fish[32]={ }的元素为鱼吐气泡图案的点阵码)

```
int CVICALLBACK start (int panel, int control, int event,
                       void *callbackData, int eventDatal, int eventData2)
                          {  unsigned char tree[32]={ }; unsigned char fish[32]={ }
  int i,j,K,speed;
 switch (event)
  {   case EVENT_COMMIT:
      GetCtrlVal (panelHandle, PANEL_SPEED, &speed);
      for (j=0; j<=speed; j++)                    //设置显示刷新时间
       for(i=0; i<16; i++)                        //点阵显示 tree
      { epp_write_data(DOTLINE,i);
        epp_write_data(DOTL,tree [i*2]);
        epp_write_data(DOTR,tree[i*2+1]);
         for (k=0;k<5;k++)                        //延时
          epp_write_data(DOTLINE,i);              //熄灭当前行
          epp_write_data(DOTL,0);
          epp_write_data(DOTR,0);}
       Delay(0.5);
       for(i=0; i<16; i++)                        //点阵显示 fish
      { epp_write_data(DOTLINE,i);
        epp_write_data(DOTL, fish[i*2]);
        epp_write_data(DOTR,fish[i*2+1]);
         for (k=0;k<5;k++)                        //延时
          epp_write_data(DOTLINE,i);              //熄灭当前行
          epp_write_data(DOTL,0);
          epp_write_data(DOTR,0);}
    break;
 ……}
```

3．点阵显示任意字符

（1）显示任意字符的 LabWindows/CVI 编程步骤

①打开 CVI 软件，创建一个空的工程文件。进入用户界面的编辑窗口，创建 UIR 文件，对用户界面进行设计。

参考设计界面如下图；控件的主要属性设置如下表

面板主要控件的基本属性列表

	类型	Constant Name	Callback Function	作用
面板	Panel	PANEL		
控件	Command Button	START	Start	控制显示任意字符
	Command Button	QUIT	Quit	退出程序
	Ring Slide	SPEED		显示刷新速度
	String	INPUT		输入显示字符
	Decoration	DECORATION_1—2		美观

②在用户界面编辑窗口，由图形界面生成源代码框架，并给工程文件命名。打开 SJ8002C 驱动函数文件“sj8002C.fp”并添加到工程文件中。

③ 向 C 代码文件添加函数。（调用 sj8002C.fp）

1)EPP 初始化函数 epp_init ();

无输入参数，无返回值。

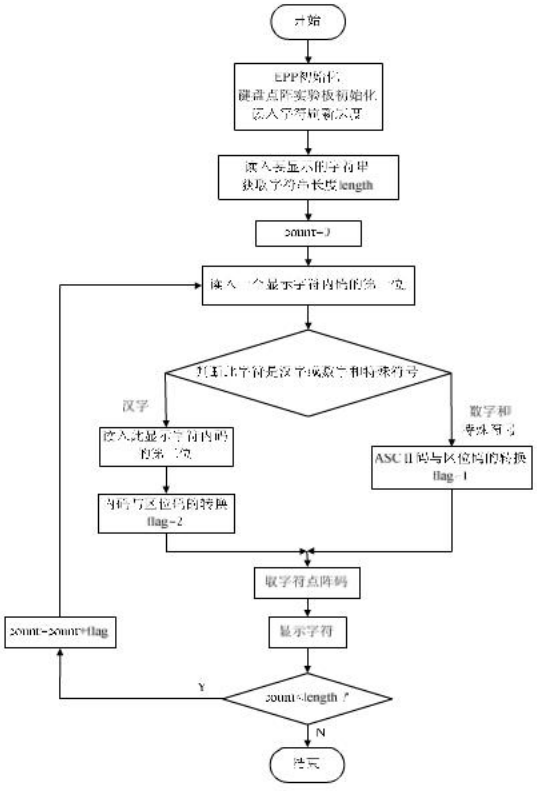
2)显示多个字符 led_dots (char *inputs, int speed)

输入值： inputs （用户面板上输入的 char 数据(可以是数字,特殊字符和汉字)）；speed（刷新速度）。

函数无返回值。

注意：必须把字库文件 ’hzk16j.dot’ 拷贝到程序运行的目录下

（2）点阵的编程思路



四．操作调试步骤

（1）硬件连线

用 20PIN 的带线连接电子测量实验箱主板 J005（20PIN 插座）和点阵实验板 J1。连通 EPP 线。

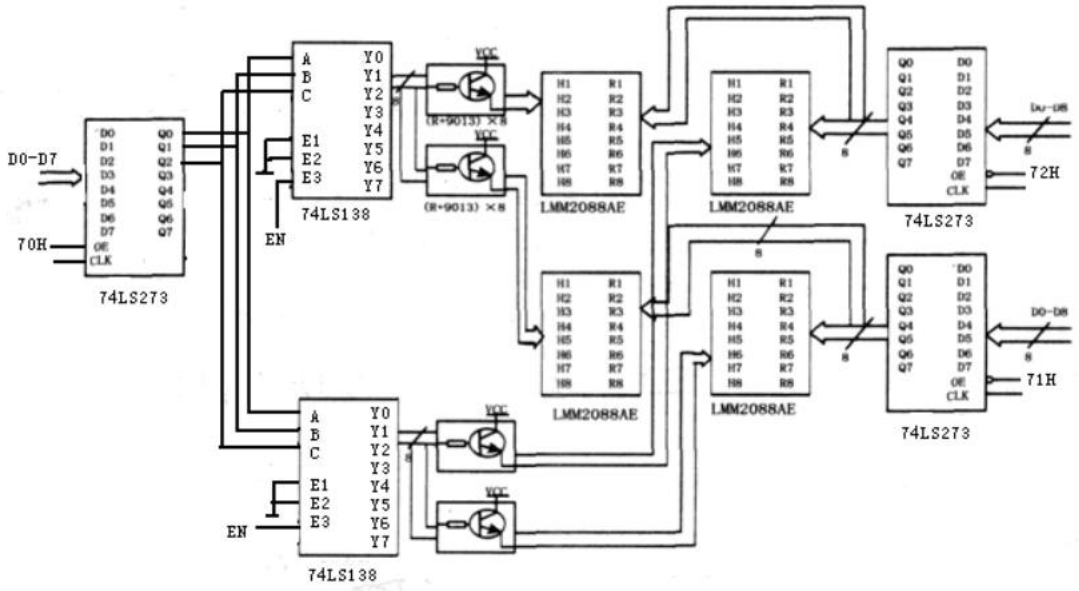
（2）点阵显示

“显示刷新速度”调整的是点阵显示刷新速度。

运行人机对话实验软件界面后，出现界面。点击“演示”，将在“字符输入”框自动的显示静态的字符“欢迎使用电子科技大学虚拟仪器电子测量实验系统”，“字符输入”框动态的显示，点阵实验板的点阵面板上逐个的显示以上的汉字。

在“字符输入”框里可以输入任意字符，再点击“显示”，输入的字符在“字符输出”框动态显示，同时，点阵面板上也个逐个显示同样的字符。

付图：点阵显示板电路



五、设计要求

1） 画出点阵显示板电路原理图（用 Protel 软件）。

2） 列出电路元器件清单。